November 1988

UILU-ENG-88-2255
CSG-91

COORDINATED SCIENCE LABORATORY
College of Engineering

Langley Grant
IN 61

PERFORMANCE ANALYSIS OF THE ALLIANT FX/8 MULTIPROCESSOR USING STATISTICAL CLUSTERING

Robert Tod Dimpsey

UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION Unclassified	1b. RESTRICTIVE MARKINGS None
2a. SECURITY CLASSIFICATION AUTHORITY	3. DISTRIBUTION/AVAILABILITY OF REPORT Approved for public release; distribution unlimited
2b. DECLASSIFICATION/DOWNGRADING SCHEDULE	
4. PERFORMING ORGANIZATION REPORT NUMBER(S) UILU-ENG-88-2255 (CSG-91)	5. MONITORING ORGANIZATION REPORT NUMBER(S)
6a. NAME OF PERFORMING ORGANIZATION Coordinated Science Lab University of Illinois — 6b. OFFICE SYMBOL (If applicable) N/A	7a. NAME OF MONITORING ORGANIZATION NASA
6c. ADDRESS (City, State, and ZIP Code) 1101 W. Springfield Ave. Urbana, IL 61801	7b. ADDRESS (City, State, and ZIP Code) NASA Langley Research Center Hampton, VA 23665
8a. NAME OF FUNDING/SPONSORING ORGANIZATION NASA — 8b. OFFICE SYMBOL (If applicable) N/A	9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER NASA-NAG-1-613
8c. ADDRESS (City, State, and ZIP Code) See block 7b.	10. SOURCE OF FUNDING NUMBERS PROGRAM ELEMENT NO. / PROJECT NO. / TASK NO. / WORK UNIT ACCESSION NO.

11. TITLE (Include Security Classification)
Performance Analysis of the Alliant FX/8 Multiprocessor Using Statistical Clustering

12. PERSONAL AUTHOR(S)
Dimpsey, Robert Tod

13a. TYPE OF REPORT	13b. TIME COVERED	14. DATE OF REPORT (Year, Month, Day)	15. PAGE COUNT
Technical	FROM ______ TO ______	1988 November	36

16. SUPPLEMENTARY NOTATION

17. COSATI CODES

FIELD	GROUP	SUB-GROUP

18. SUBJECT TERMS (Continue on reverse if necessary and identify by block number)
statistical modeling, statistical clustering, resource usage performance analysis, shared memory multiprocessor, skewness factor, workload modeling, concurrency, state trans. model

19. ABSTRACT (Continue on reverse if necessary and identify by block number)

This paper presents an analysis of an Alliant FX/8 system running Xylem (Cedar's operating system) at the University of Illinois Center for Supercomputing Research and Development. Results for two distinct, real, scientific workloads executing on an Alliant FX/8 are discussed. A combination of user concurrency and system overhead measurements was taken for both workloads. Preliminary analysis shows that the first sampled workload is comprised of consistently high user concurrency, low system overhead, and little paging. The

20. DISTRIBUTION/AVAILABILITY OF ABSTRACT [✓ UNCLASSIFIED/UNLIMITED] ☐ SAME AS RPT. ☐ DTIC USERS	21. ABSTRACT SECURITY CLASSIFICATION Unclassified
22a. NAME OF RESPONSIBLE INDIVIDUAL	22b. TELEPHONE (Include Area Code) — 22c. OFFICE SYMBOL

DD Form 1473, JUN 86 *Previous editions are obsolete.* SECURITY CLASSIFICATION OF THIS PAGE

UNCLASSIFIED

ORIGINAL PAGE IS
OF POOR QUALITY

second sample has much less user concurrency, but significant paging and system overhead.

Statistical cluster analysis is used to extract a state transition model to jointly characterize user concurrency and system overhead. A skewness factor is introduced and used to bring out the effects of unbalanced clustering when determining states with significant transitions.

The results from the models show that during the collection of the first sample, the system was operating in states of high user concurrency approximately 75% of the time. The second workload sample shows the system in high user concurrency states only 26% of the time. In addition, it is ascertained that high system overhead is usually accompanied by low user concurrency. The analysis also shows a high predicatability of system behavior for both workloads. This predictability is largely due to slow changes in system states. In particular, states with extremely high values of paging or user concurrency are usually preceded by states with less paging and user concurrency, much like stair climbing. The opposite effect is observed when the machine leaves these extreme states.

PERFORMANCE ANALYSIS OF THE ALLIANT FX/8 MULTIPROCESSOR USING STATISTICAL CLUSTERING

BY

ROBERT TOD DIMPSEY

B.S., University of Illinois, 1986

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master of Science in Electrical Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 1988

Urbana, Illinois

This research was supported by the **National Aeronautics and Space Administration** (NASA) under Contract NASA NAG 1-613 in cooperation with the Illinois Computer Laboratory for Aerospace Systems and Software (ICLASS), a NASA-supported Center for Excellence.

ABSTRACT

This paper presents an analysis of an Alliant FX/8 system running Xylem (Cedar's operating system) at the University of Illinois Center for Supercomputing Research and Development. Results for two distinct, real, scientific workloads executing on an Alliant FX/8 are discussed. A combination of user concurrency and system overhead measurements was taken for both workloads. Preliminary analysis shows that the first sampled workload is comprised of consistently high user concurrency, low system overhead, and little paging. The second sample has much less user concurrency, but significant paging and system overhead.

Statistical cluster analysis is used to extract a state transition model to jointly characterize user concurrency and system overhead. A skewness factor is introduced and used to bring out the effects of unbalanced clustering when determining states with significant transitions.

The results from the models show that during the collection of the first sample, the system was operating in states of high user concurrency approximately 75% of the time. The second workload sample shows the system in high user concurrency states only 26% of the time. In addition, it is ascertained that high system overhead is usually accompanied by low user concurrency. The analysis also shows a high predicatability of system behavior for both workloads. This predictability is largely due to slow changes in system states. In particular, states with extremely high values of paging or user concurrency are usually preceded by states with less paging and user concurrency, much like stair climbing. The opposite effect is observed when the machine leaves these extreme states.

ACKNOWLEDGMENTS

The author would like to thank the researchers at the Center for Supercomputing Research and Development for their assistance during the course of this work. In particular, thanks are due to Richard Barton and Bob McGrath for use of their software and to Allen Maloney for many useful discussions. Thanks are also due to Sujatha Subramani for her assistance with this workload and to Janet Adams for her careful proofreading. Special thanks go to Professor Ravi Iyer for his extremely helpful advice in regard to this thesis.

TABLE OF CONTENTS

CHAPTER ONE
INTRODUCTION

The evaluation of a computer system, in particular, a multiprocessor, is an important step in optimizing the design of the system and improving application programs for it. Too often in the field of parallel processing this evaluation consists of determining numerical performance indices, such as MFLOPS, for the machine executing a standard benchmark. Although these indices are useful in detecting global weaknesses in the system, they are unable to provide a detailed analysis. This type of evaluation is also unable to indicate how the machine will perform in the local environment. It is useful to have methods which provide information about the system's performance under a certain workload, along with insight into how the workload and system are interacting. With such methods, the system can be more easily tuned for specific applications and vice versa.

This thesis presents an analysis of an Alliant FX/8 system running the Cedar[1] operating system, Xylem, at the University of Illinois Center for Supercomputing Research and Development. Results for two distinct, real, scientific workloads executing on an Alliant FX/8 are presented. For this evaluation, a combination of user concurrency and system overhead (e.g., paging, and context switches) measurements are collected. Statistical clustering is performed on these measurements to identify commonly recurring patterns of resource usage. State transition models are extracted and interpreted for both sampled workloads to obtain practical insight into the system behavior. Skewness factors are then calculated for each interstate transition in the identified model and used to determine significant transitional relationships among the states of the machine.

The results show that during the collection of the first sample, the system was operating in states of high user concurrency approximately 75% of the time. The second sample,

[1]The Cedar project is a parallel supercomputing experiment which consists of interconnecting Alliant FX/8's to a large shared global memory [1], [2], and [3]. Each Alliant is known as a cluster of the Cedar machine. The first version of Cedar will consist of two clusters.

on the other hand, captures a system operating in states of high user concurrency only 26% of the time. In addition, it was discovered that high system overhead is usually accompanied by low user concurrency. The analysis also indicates that for both workloads, the state of the system was highly predictable. This predictability was largely due to slow changes in system states. In particular, states with extremely high values of paging or user concurrency are usually preceded by states with less paging and user concurrency, much like stair climbing. The opposite effect is observed when the machine leaves these extreme states.

1.1. Related Research

There have been many performance and concurrency studies performed on multiprocessor systems. Most of these have employed simulation and analytical-based techniques [4], [5], [6], [7], without investigating the effect of a real workload. There have also been a few performance evaluation studies done on the Alliant machine [8], [9], [10], and [11]. Most of these are concerned with the use of tools for evaluation. For instance, McGuire and Iyer [8] instrument the Alliant FX/8 to measure concurrency present in a real workload. In their work, total concurrency is measured ignoring the distinction between user and system related concurrency. Measurements of user concurrency alone are particularly important if the evaluation is being done as a step in optimizing an application program. For this reason, this study primarily deals with user concurrency.

Unlike the above studies, this study does not only pursue performance indices for the system but also extracts models of the executing workloads. Very little work has been done in this area for multiprocessor systems. This thesis presents two case studies in identifying useful models of real workloads on a multiprocessor. These models are also interpreted to gain insight into the interaction of the workload and system and to determine the amount of concurrency in the workloads.

A major step in obtaining the workload models is statistical clustering. In recent years, statistical clustering has found many uses in the field of computer evaluation. Devarakonda and Iyer [12] use clustering as a step in creating transition models which are then used to predict resource usage. Hsueh et al. [13] use similar techniques to create performability models for a multiprocessor system. Ferrari [14], on the other hand, uses clustering in the creation of artificial workloads.

The next chapter contains a discussion of the measured environment. Chapter 3 introduces the measurements used in this study and explains how they were obtained. A number of preliminary results for the collected samples are presented in Chapter 4. These provide a better understanding of the two workloads sampled and their interaction with the machine. Following this in Section 5.1 is a discussion of the clustering procedure and the method used to create the transition models. In addition, this section introduces the skewness factor, provides its definition, and discusses the reasons for it. The identified workload models and thorough interpretations of them can be found in Sections 5.2 and 5.3. Chapter 6 summarizes the major results and suggests possibilities for future work.

CHAPTER TWO

THE MEASUREMENT ENVIRONMENT

The measurements for this study were taken from real, scientific workloads being executed by an Alliant FX/8 on weekday afternoons. The FX/8 is a multiprocessor mini-supercomputer with a 32 Megabyte shared global memory [15]. It can best be understood as two groups or clusters[2] of processors. The main group, the Computational Element (CE) cluster, consists of eight processors. These either work together concurrently in the clustered configuration or separately in the detached configuration. When the CEs are detached, they can be used as eight separate processors working on different jobs, or groups of them can be used to multiprocess the same job. When in the clustered configuration, the concurrency control bus synchronizes the eight CEs to concurrently process a single job.

The second group of processors on the measured Alliant consists of three Motorola MC68012 microprocessors called the Interactive Processors (IPs). For the situation being studied, the IPs handle all accesses to secondary memory and interactive user work such as editing jobs. It is important to note that the operating system on the measured machine is Xylem, which was specifically designed for the Cedar supercomputer, and not Concentrix, Alliant's operating system. For this reason, this thesis is more correctly viewed as an analysis of a single cluster Cedar supercomputer, and not as an analysis of the Alliant FX/8. This distinction may seem slight, but it becomes important later in this thesis (Chapter 4).

The measured FX/8 is used for application and algorithm development at the University of Illinois' Center for Supercomputer Research and Development (UICSRD). Work being done on the machine varies from the creation of a mathematical library containing optimal versions of commonly used algorithms such as fast Fourier transforms and

[2]The use of the word cluster is admittedly overused in this thesis. The Alliant FX/8s are clusters of the Cedar, while the FX/8's have their own clusters. Later, cluster models will be introduced. This confusion was inevitable, in order to maintain consistency with the results in the other literature on these subjects.

solutions of sparse linear systems, to the development of a highly concurrent circuit simulator. In addition, the newly developed Cedar operating system Xylem was being tested and debugged during the course of this work. This diverse environment allowed us to measure programs specifically designed to optimize the concurrency allowed by Cedar's architecture along with jobs that were suboptimal. In general, the measured workload is representative of many scientific, parallel program developmental situations.

CHAPTER THREE

MEASUREMENTS

Two software facilities developed at UICSRD were used to measure system behavior. The first was used to measure concurrency exploited on the CE cluster, and the second was used to measure system related overhead. These facilities monitored the system concurrently so both types of measurements were collected simultaneously.

To determine the amount of concurrency in the workload, a software program using a high resolution (10 microsecond) timer measured the amount of time each processor was executing system and user code, as well as the amount of time each processor was idle. These measurements were taken separately for the two CE configurations (i.e., detached and clustered). In this way, the percentage of time the CEs were clustered and executing user code (CONCUSER) was determined. The CONCUSER parameter directly corresponds to the amount of user concurrency in the workload and will be high for observations with well-tuned applications running.[3]

System overhead was measured using an operating system facility, which monitors and collects data on virtual memory and system operations such as paging, swapping, system calls, context switches, and file searches. Of the approximately 150 measurements

Table 1
Virtual Memory Measurements

Variable	Description
context switch	preemption of currently running program
device interrupt	access to devices
page in	access to disk to bring pages into main memory
page out	access to disk to write pages out
pages paged in	number of pages brought into main memory
pages paged out	number of pages written to the disk

[3] For this study the time spent in detached configuration with more than one CE executing user code is not considered concurrent operation. This happens infrequently (as will be seen in the next chapter), and usually on two separate programs.

available, those sampled for this study are summarized in Table 1. It should be noted that the O/S facility does not provide separate measurements for each processor, but running totals for all the processors combined. For instance, the context switch measurement contains the total number of context switches that have occurred on all eight CEs plus the three IPs.

In addition to these measurements, the parameters summarized in Table 2 were calculated. Notice that some of the percentages in the table are calculated over the entire observation period, and others are calculated just over the time spent in a specific configuration. The parameters CEUT and IPUT refer to the utilization of the entire CE and IP complex, respectively, and are defined as follows:

$$CEUT = \frac{\sum_{j=0}^{7} CEj\ user\ time + \sum_{j=0}^{7} CEj\ system\ time + 8^{*}\{(cluster\ user\ time) + (cluster\ system\ time)\}}{(number\ of\ CEs)^{*}\ (sampling\ period)}$$

$$IPUT = \frac{\sum_{j=0}^{2} IPj\ user\ time + \sum_{j=0}^{2} IPj\ system\ time}{(number\ of\ IPs)^{*}\ (sampling\ period)}$$

Table 2
Utilization Measurements

Measure	Description
CONCUSER	% of time CEs clustered and running user code
clsyst	% of cluster time spent running system code
cluset	% of cluster time spent running user code
CLUSTIM	% of time spent in the cluster configuration
ipsyst	% of time IPs spent running system code
CEUT	CE utilization
IPUT	IP utilization

All measurements discussed above were sampled simultaneously every 45 seconds.[4] Each 45-second period is one observation of the system, and the measurements collected during that period depict the state of the system for that observation. The length of the observation was experimentally determined and chosen so that it would best correspond to the length of an actual, physical state of the machine.

The observation length was initially tuned with respect to paging since this was the most elusive parameter. Early measurements indicated paging activity most often came in ninety-second blocks surrounded by periods of few disk accesses. The 45-second (90 divided by 2) length was then chosen so that paging activity could be captured without losing the majority of the head or tail of its block to the neighboring observations. Ideally then, a paging block would be captured as two 45-second high paging observations. The 45-second interval worked well with the other parameters, so it was kept as the period of time defining a state for this machine and workload environment.

Many samples of the Alliant workload at UICSRD have been collected and studied. Each sample is generally two to three hours long.[5] In this thesis, two markedly different samples are presented. The first sample was taken over a 138-minute period. The second sample, on the other hand, is 168 minutes long. To provide a broad understanding of the two workloads and their interactions with the system, some preliminary statistical analysis is presented in the next chapter.

[4]I am using two separate measuring facilities so the measurements were not sampled exactly simultaneously, but sequentially. The VM facility sampled its measurements, and this was immediately followed by the concurrency facility. The gap between the sampling was of little consequence to this work because it was very small in relation to the 45-second sampling interval.

[5]Larger samples would have been collected but continuous periods of substantial workload without a system failure were fairly short (2-3 hours). This was due to the operating system still being in its infancy and therefore not being completely installed or debugged.

CHAPTER FOUR

PRELIMINARY ANALYSIS

4.1. Means and Standard Deviations

Table 3 contains observation means and standard deviations for each parameter studied. The results for Sample One show that the CE complex was consistently well utilized (CEUT mean = 0.723, std. dev. = 0.077). On the other hand, the low mean (0.393) and high standard deviation (0.246) shown for the CE utilization of Sample Two implies either an ill-tuned system or a workload consisting of bursts of work surrounded by idleness. The cluster model (Chapter 5) will address this in more detail.

High standard deviations are seen for all measurements collected for Sample Two. This indicates that the sample captured a range of different activity. In contrast, the low standard deviations of the parameters for Sample One indicate that the captured workload had consistent behavior throughout the sampling.

A closer examination of Table 3 shows the imbalance between the mean IP and mean CE utilizations: 0.304 to 0.723 for Sample One, and 0.271 to 0.393 for Sample Two. The low standard deviations for the IP utilizations indicate that the underutilization of the IPs and the imbalance of work between the processor complexes is fairly steady throughout the observations. This imbalance, especially for Sample One, may be partially attributed to the low paging activity. (All accesses to disk must be made through IP0, which would cause the utilization of the IP complex to increase and reduce this imbalance.) A second cause of this imbalance is rooted in the switch from the Concentrix to the Xylem operating system.[6] Based on these results, it seems appropriate to offload some work from the CEs to the IPs. More study is necessary to quantify the benefit, if any, of such a transfer.

[6]A good portion of Xylem is executed by the CEs. This is done because only the CEs can access the Cedar global memory. The Alliant, on the other hand, was designed to execute a good deal of the O/S with the IPs. Therefore, if the same workload was run under Concentrix, the IP/CE imbalance would lessen.

Table 3
Parameter Means and Standard Deviations

Parameter	Sample One		Sample Two	
	mean	std. dev.	mean	std. dev.
context switches	1782.508	508.201	1503.382	665.230
device interrupts	20389.339	5976.630	18459.958	11929.022
page ins	0.109	1.016	24.747	66.278
pages paged in	35.880	112.737	159.089	360.887
page outs	1.869	14.630	18.116	40.957
pages paged out	34.038	269.249	339.938	770.356
CE utilization	0.723	0.077	0.393	0.246
IP utilization	0.304	0.078	0.271	0.101
CLUSTIM	71.632	10.472	63.920	14.971
cluset	90.165	4.946	39.879	34.000
clsyst	9.835	4.946	21.727	15.159
ipsyst	23.716	5.325	17.231	4.146
CONCUSER	64.625	10.470	27.028	26.028

Table 3 clearly shows the paging differences between the two sampled workloads. The first sample contained on average 0.109 accesses to the disk for pages being brought into memory and 1.869 accesses for pages written to the disk per observation. In contrast, Sample Two contained an average of 24.7 accesses to the disk for pages ins and an average of 18.1 accesses for page outs per observation. This cannot be considered a large amount of paging, but it is certainly enough to have a significant effect on the behavior of the system. The standard deviations for the paging activities of both samples are quite high, suggesting intervals of high paging activity in addition to long periods of little or no paging. The long periods of little paging activity are easily explained by the large 32-KB physical memory found in the Alliant.

As mentioned earlier, the parameter corresponding to user concurrency is CONCUSER. It is obvious from Table 3 that Sample One captured a good deal more user concurrency than Sample Two (mean Sample One CONCUSER = 65%, mean Sample Two CONCUSER = 26%). It is interesting to note that although the amount of user concurrency in the two samples is drastically different, the percentage of time spent in the clustered configuration (CLUSTIM) is very similar (CLUSTIM mean = 71% and 64%, CLUSTIM std. dev. = 10 and

15) for the two samples. An explanation of this phenomenon is discussed in the next section.

4.2. Individual System/User/Idle Times

For the majority of this thesis, the CE and IP complexes are treated as indivisible units. In this section though, the behavior of the individual processors is studied. Tables 4 and 5 show the percentage of time each processor spends executing system and user code, along with the percentage of time the processors are idle. The bars shown for the individual CEs, CE0–CE7, pertain to the time spent in detached configuration. The cluster bar (CL) shows the breakdown for the CEs' utilizations while in the clustered configuration (only one bar is needed because all CEs work on the same job in this configuration). It is important to realize that these percentages are not calculated over the whole period, but only the period in which the CEs are in the specified configuration. For example, Table 4 shows that while detached, CE7 is idle 45% of the time, executing system code 30% of the time, and executing user code 25% of the time.

Tables 4 and 5 provide the explanation to the question left unanswered at the end of the last section. The difference in user concurrency exhibited in the two samples is not explained by the percentage of time the CEs were clustered, but by the type of work they were doing while clustered. While the machine was clustered during the collection of Sample One, the CEs were executing user code about 90% of the time and system code about 10% of the time. The cluster was never idle. Sample Two, on the other hand, captures an idle cluster about 39% of the time. This idleness while clustered is the cause of the drastic difference in the amount of user concurrency observed in the two samples.

Tables 4 and 5 confirm the underutilization of the IPs. They also show that the work done on the IPs is evenly balanced. Another point of interest on these tables is the low utilization of the CEs while in the detached mode. CE0 and CE1 are never used; CE2 is used

ORIGINAL PAGE IS
OF POOR QUALITY

Table 4
System/User/Idle Times
Sample One

```
1.0 +  IIII  IIII  UUUU  UUUU  UUUU  UUUU  UUUU  UUUU  UUUU  SSSS  UUUU  UUUU
    :  IIII  IIII  IIII  SSSS  UUUU  UUUU  UUUU  UUUU  UUUU  SSSS  UUUU  UUUU
    :  IIII  IIII  IIII  IIII  UUUU  UUUU  UUUU  UUUU  UUUU  SSSS  UUUU  UUUU
    :  IIII  IIII  IIII  IIII  SSSS  UUUU  UUUU  UUUU  UUUU  SSSS  SSSS  UUUU
0.9 +  IIII  IIII  IIII  IIII  SSSS  UUUU  SSSS  UUUU  UUUU  SSSS  SSSS  UUUU
    :  IIII  IIII  IIII  IIII  IIII  UUUU  SSSS  UUUU  UUUU  SSSS  SSSS  SSSS
    :  IIII  IIII  IIII  IIII  IIII  UUUU  SSSS  UUUU  UUUU  SSSS  SSSS  SSSS
    :  IIII  IIII  IIII  IIII  IIII  UUUU  SSSS  UUUU  UUUU  SSSS  SSSS  SSSS
0.8 +  IIII  IIII  IIII  IIII  IIII  UUUU  SSSS  UUUU  UUUU  SSSS  SSSS  SSSS
    :  IIII  IIII  IIII  IIII  IIII  SSSS  SSSS  UUUU  UUUU  SSSS  IIII  SSSS
    :  IIII  IIII  IIII  IIII  IIII  SSSS  SSSS  SSSS  UUUU  SSSS  IIII  SSSS
    :  IIII  IIII  IIII  IIII  IIII  SSSS  IIII  SSSS  UUUU  SSSS  IIII  SSSS
0.7 +  IIII  IIII  IIII  IIII  IIII  SSSS  IIII  SSSS  UUUU  IIII  IIII  SSSS
    :  IIII  IIII  IIII  IIII  IIII  IIII  IIII  SSSS  UUUU  IIII  IIII  SSSS
    :  IIII  IIII  IIII  IIII  IIII  IIII  IIII  SSSS  UUUU  IIII  IIII  SSSS
    :  IIII  IIII  IIII  IIII  IIII  IIII  IIII  SSSS  UUUU  IIII  IIII  IIII
0.6 +  IIII  IIII  IIII  IIII  IIII  IIII  IIII  SSSS  UUUU  IIII  IIII  IIII
    :  IIII  IIII  IIII  IIII  IIII  IIII  IIII  SSSS  UUUU  IIII  IIII  IIII
    :  IIII  IIII  IIII  IIII  IIII  IIII  IIII  SSSS  UUUU  IIII  IIII  IIII
    :  IIII  IIII  IIII  IIII  IIII  IIII  IIII  SSSS  UUUU  IIII  IIII  IIII
0.5 +  IIII  IIII  IIII  IIII  IIII  IIII  IIII  SSSS  UUUU  IIII  IIII  IIII
    :  IIII  IIII  IIII  IIII  IIII  IIII  IIII  SSSS  UUUU  IIII  IIII  IIII
    :  IIII  IIII  IIII  IIII  IIII  IIII  IIII  IIII  UUUU  IIII  IIII  IIII
    :  IIII  IIII  IIII  IIII  IIII  IIII  IIII  IIII  UUUU  IIII  IIII  IIII
0.4 +  IIII  IIII  IIII  IIII  IIII  IIII  IIII  IIII  UUUU  IIII  IIII  IIII
    :  IIII  IIII  IIII  IIII  IIII  IIII  IIII  IIII  UUUU  IIII  IIII  IIII
    :  IIII  IIII  IIII  IIII  IIII  IIII  IIII  IIII  UUUU  IIII  IIII  IIII
    :  IIII  IIII  IIII  IIII  IIII  IIII  IIII  IIII  UUUU  IIII  IIII  IIII
0.3 +  IIII  IIII  IIII  IIII  IIII  IIII  IIII  IIII  UUUU  IIII  IIII  IIII
    :  IIII  IIII  IIII  IIII  IIII  IIII  IIII  IIII  UUUU  IIII  IIII  IIII
    :  IIII  IIII  IIII  IIII  IIII  IIII  IIII  IIII  UUUU  IIII  IIII  IIII
    :  IIII  IIII  IIII  IIII  IIII  IIII  IIII  IIII  UUUU  IIII  IIII  IIII
0.2 +  IIII  IIII  IIII  IIII  IIII  IIII  IIII  IIII  UUUU  IIII  IIII  IIII
    :  IIII  IIII  IIII  IIII  IIII  IIII  IIII  IIII  UUUU  IIII  IIII  IIII
    :  IIII  IIII  IIII  IIII  IIII  IIII  IIII  IIII  UUUU  IIII  IIII  IIII
    :  IIII  IIII  IIII  IIII  IIII  IIII  IIII  IIII  UUUU  IIII  IIII  IIII
0.1 +  IIII  IIII  IIII  IIII  IIII  IIII  IIII  IIII  SSSS  IIII  IIII  IIII
    :  IIII  IIII  IIII  IIII  IIII  IIII  IIII  IIII  SSSS  IIII  IIII  IIII
    :  IIII  IIII  IIII  IIII  IIII  IIII  IIII  IIII  SSSS  IIII  IIII  IIII
    :  IIII  IIII  IIII  IIII  IIII  IIII  IIII  IIII  SSSS  IIII  IIII  IIII
    ----------------------------------------------------------------------------
       CE0   CE1   CE2   CE3   CE4   CE5   CE6   CE7   CL    IP0   IP1   IP2
```

Computational Elements — Interactive Processors

Percentage of Time CEs in clustered configuration: 72%

U - User

S - System

I - Idle

Table 5
System/User/Idle Times
Sample Two

```
1.0 + IIII IIII IIII SSSS UUUU UUUU UUUU UUUU UUUU SSSS UUUU UUUU
    : IIII IIII IIII SSSS SSSS UUUU UUUU UUUU UUUU SSSS SSSS UUUU
    : IIII IIII IIII IIII SSSS UUUU UUUU UUUU UUUU SSSS SSSS UUUU
    : IIII IIII IIII IIII IIII UUUU UUUU UUUU UUUU SSSS SSSS SSSS
0.9 + IIII IIII IIII IIII IIII SSSS SSSS UUUU UUUU SSSS SSSS SSSS
    : IIII IIII IIII IIII IIII SSSS SSSS UUUU UUUU SSSS SSSS SSSS
    : IIII IIII IIII IIII IIII SSSS SSSS UUUU UUUU SSSS SSSS SSSS
    : IIII IIII IIII IIII IIII SSSS SSSS UUUU UUUU SSSS IIII SSSS
0.8 + IIII IIII IIII IIII IIII IIII SSSS UUUU UUUU SSSS IIII SSSS
    : IIII IIII IIII IIII IIII IIII SSSS UUUU UUUU SSSS IIII SSSS
    : IIII IIII IIII IIII IIII IIII SSSS SSSS UUUU SSSS IIII SSSS
    : IIII IIII IIII IIII IIII IIII IIII SSSS UUUU SSSS IIII SSSS
0.7 + IIII IIII IIII IIII IIII IIII IIII SSSS UUUU SSSS IIII SSSS
    : IIII IIII IIII IIII IIII IIII IIII SSSS UUUU SSSS IIII IIII
    : IIII IIII IIII IIII IIII IIII IIII SSSS UUUU IIII IIII IIII
    : IIII IIII IIII IIII IIII IIII IIII SSSS UUUU IIII IIII IIII
0.6 + IIII IIII IIII IIII IIII IIII IIII SSSS UUUU IIII IIII IIII
    : IIII IIII IIII IIII IIII IIII IIII SSSS UUUU IIII IIII IIII
    : IIII IIII IIII IIII IIII IIII IIII SSSS SSSS IIII IIII IIII
    : IIII IIII IIII IIII IIII IIII IIII SSSS SSSS IIII IIII IIII
0.5 + IIII IIII IIII IIII IIII IIII IIII SSSS SSSS IIII IIII IIII
    : IIII IIII IIII IIII IIII IIII IIII SSSS SSSS IIII IIII IIII
    : IIII IIII IIII IIII IIII IIII IIII IIII SSSS IIII IIII IIII
    : IIII IIII IIII IIII IIII IIII IIII IIII SSSS IIII IIII IIII
0.4 + IIII IIII IIII IIII IIII IIII IIII IIII SSSS IIII IIII IIII
    : IIII IIII IIII IIII IIII IIII IIII IIII SSSS IIII IIII IIII
    : IIII IIII IIII IIII IIII IIII IIII IIII SSSS IIII IIII IIII
    : IIII IIII IIII IIII IIII IIII IIII IIII IIII IIII IIII IIII
0.3 + IIII IIII IIII IIII IIII IIII IIII IIII IIII IIII IIII IIII
    : IIII IIII IIII IIII IIII IIII IIII IIII IIII IIII IIII IIII
    : IIII IIII IIII IIII IIII IIII IIII IIII IIII IIII IIII IIII
    : IIII IIII IIII IIII IIII IIII IIII IIII IIII IIII IIII IIII
0.2 + IIII IIII IIII IIII IIII IIII IIII IIII IIII IIII IIII IIII
    : IIII IIII IIII IIII IIII IIII IIII IIII IIII IIII IIII IIII
    : IIII IIII IIII IIII IIII IIII IIII IIII IIII IIII IIII IIII
    : IIII IIII IIII IIII IIII IIII IIII IIII IIII IIII IIII IIII
0.1 + IIII IIII IIII IIII IIII IIII IIII IIII IIII IIII IIII IIII
    : IIII IIII IIII IIII IIII IIII IIII IIII IIII IIII IIII IIII
    : IIII IIII IIII IIII IIII IIII IIII IIII IIII IIII IIII IIII
    : IIII IIII IIII IIII IIII IIII IIII IIII IIII IIII IIII IIII
    -------------------------------------------------------------
      CE0  CE1  CE2  CE3  CE4  CE5  CE6  CE7  CL   IP0  IP1  IP2
```

Computational Elements — Interactive Processors

Percentage of
Time CEs
in clustered
configuration:
64%

U - User

S - System

I - Idle

only sparingly in Sample One and not at all in Sample Two, and CE3 and CE4 are only slightly used. The majority of the work in the detached mode is done by CE6 and CE7. These results suggest that a better design may be to allow the four lower CEs to form their own cluster. In this way, when the detached mode is needed, the upper four processors can break free and handle the work. Meanwhile, the lower four stay in clustered configuration and continue to service the jobs waiting on the cluster queue. With this method, the utilization of the CE complex will most likely improve.

4.3. Correlations

Correlations between all combinations of parameters were calculated to investigate the relationships and dependencies among the measurements taken. The most interesting results are presented in Table 6. The first entry shows that although page ins and page outs are highly correlated for Sample Two, they are not for Sample One. This is probably a result of the low paging activity captured in Sample One, coupled with the presence of page reclaims.

The correlations between CEUT and IPUT are also very interesting because they lead to completely opposite conclusions for the two samples. The positive correlation for Sample Two (0.6266) suggests that the IP and CE utilizations follow the same general pattern. They are both high or both low for the same intervals of time. This indicates that the sample captured a generally light workload which contained scattered periods of high activity.

Table 6
Selected Correlations

Parameters		Sample One Correlation	Sample Two Correlation
page ins	page outs	-0.01382	0.66765
CEUT	IPUT	-0.79943	0.62660
context switches	ipsyst	0.91161	0.78697
device interrupts	clsyst	0.89835	0.69898
CEUT	CLUSTIM	0.96672	0.38059

During these high activity periods, both the CE and IP utilizations increased, resulting in the high correlation. The negative correlation for Sample One (-0.77184), on the other hand, indicates that when the CEs are busy, the IPs are not, and vice versa. This is probably caused by the system's consistent operation in one of two states. In the first state, the machine spends the majority of its time with the CEs in the clustered configuration executing user code concurrently (high CEUT, low IPUT). This, of course, is the desirable state. The other state consists of a good deal of system work being executed by the machine causing the CEs to spend more time detached (low CEUT, high IPUT).

The correlation between ipsyst and context switches was high for both samples, indicating that an increase in context switches (which is probably caused by an increase in the amount of multiprogramming) is generally accompanied by an increase in system activities that must be performed by the IPs. The correlation between device interrupts and clsyst was likewise found to be high for both samples. Interestingly, a similarly high correlation was not found between device interrupts and ipsyst. This means that an increase in device interrupts is generally accompanied by system activity which needs to be executed on the cluster, but not on the IPs.

The correlations between CEUT and CLUSTIM were included to show that although at times the percentage of time the CEs spend clustered is highly related to the CE utilization, this is not always the case.

4.4. Summary of Preliminary Analysis

In summary, the preliminary analysis shows that Sample One captured a system with high, steady CE utilization, little paging, and a good deal of user concurrency. This is the result of a relatively unchanging workload. Sample Two, on the other hand, is made up of observations with high variability in their CE utilization, and the amount of paging they capture. On average, the sample also shows very little user concurrency. This is the result of a generally light workload with bursts of high activity. In addition, it was discovered that during the collection of both samples, the lower numbered CEs in the detached configuration and all three IPs were underutilized.

CHAPTER FIVE

MODEL EXTRACTION

In this chapter state transition models are extracted to quantify the variation in system activity for each workload. Four parameters were selected to jointly characterize user concurrency and system overhead. These were IPUT, context switches, CONCUSER, and pagact (pagact = page ins + page outs, the total number of accesses to disk). Each observation is essentially a point in four-dimensional space. Statistical clustering analysis is used to identify similar classes (clusters) in this space. Each cluster is then used to depict a system state, and a transition model (consisting of intercluster transition probabilities) is developed. Following this, skewness factors are calculated and used to detect significant transitional relationships between the states of the system. Before all this is done, more detailed descriptions of the above methods are presented.

5.1. Clustering, State Transition Models, and Skewness Factors

The cluster models were obtained using the FASTCLUS procedure from the SAS software package ([16], [17], and [18]). This procedure uses a K-means clustering method, grouping observations into clusters that minimize the intracluster distances between points, while maximizing the intercluster distances. The algorithm first chooses a set of K seeds (K is the number of clusters the user has specified), and uses these to create the first iteration of clusters by grouping each observation with the seed that is nearest to it. (Euclidean distances are used to determine the closeness of points.) The means for each of the initial clusters are then used as seeds to create the second iteration of clusters. The number of iterations to be performed can be chosen by the user, but usually only two or three are needed to obtain adequate cluster models.

The collected data were normalized so that each measurement had a mean of zero and a standard deviation of one. This was done so that parameters with the largest range of values did not dominate the clustering procedure.

In this type of modeling, it is often advisable to exclude points with extreme values when forming the initial clusters. These excluded points are then added to the nearest cluster once the initial model is constructed. In this way, these outliers cannot form their own clusters of just two or three members. In this study, this technique was used sparingly. Only two observations were excluded from the initial clustering of Sample One, and no observations were omitted from Sample Two. Few observations were discarded because in order to let the data define the clusters, not force the data into clusters.

The cluster models obtained are studied from three different perspectives, each providing different types of results. Each was then useful for different applications. At the most basic level, the clusterings of observations are studied verbatim to determine the characteristics of the different states in which the machine is found. By the number of observations in each cluster, the percentage of time the machine is in each of these states may be determined. From this, the efficiency of the machine may be ascertained.

The second form of analysis requires the creation of a state transition model, which consists of the probabilities for each intercluster (interstate) transition. These probabilities are easily estimated from the collected data with the following formula (P_{ij} is the probability of transition from state *i* to state *j*):

$$P_{ij} = \frac{\textit{observed number of transitions from state i to state j}}{\textit{observed number of transitions from state i}}$$

These transition probabilities are used to predict forthcoming states of the machine. They provide a solid understanding of the relationships between states.

The final method used to interpret the extracted cluster model is my own technique which consists of computing skewness factors for each transition. These skewness factors quantify the degree to which transitional relations between states were caused by random transitions. More specifically, the skewness factor determines the skewness of a transition probability with respect to the transition probability that would be obtained if each inter-observation transition was equally likely. The skewness factor (S_{ij}) of a transition from state i to state j is defined as

$$S_{ij} = \frac{\textit{observed number of transitions from state i to state j}}{\textit{probable* number of transitions from state i to state j}}$$

*Assuming that the transition to any observation is equally likely regardless of the cluster it is in.

The skewness factors bring out the effect of unbalanced clusters and quantifies significant transitions between clusters. A significant transition is one that may have underlying system-related cause, and is not just the result of random action. A skewness factor near unity indicates that there is probably not a significant transition between states.

The usefulness of the skewness factor can be best illustrated with an example. Suppose a cluster model of 1000 observations was created with cluster A containing 40 observations, cluster B containing 450, and the rest of the observations spread among the other clusters. Now suppose 30 transitions were observed from cluster B to cluster A. The transition probability for this transition would be 0.0667, (30/450). This makes the transition appear insignificant. In contrast, the skewness factor is 1.665, $\left(\frac{30}{450 * \frac{40}{999}}\right)$, which indicates that the transition from cluster A to cluster B is a significant transition. In other words, the transition cannot be attributed to random activity. Upon further inspection it becomes clear that the skewness factor has indeed pinpointed a noteworthy transition. Notice that 75% of all transitions into state A come from state B. There is more than likely a system-related reason for this. This transition is even more noteworthy if cluster A represents an extremely desirable or undesirable state, because it provides information on the probable state the machine is operating in before the targeted state is reached. This information would be extremely useful in the tuning of a system to an application, or vice versa.

In the next example, the skewness factor is shown to determine that a transition probability that appears to point to a relation between states, may easily have been the result of random transitions. Assume 1000 observations were clustered in three clusters as shown in Figure 1. Cluster i contains 100 observations, cluster j contains 300 observations, and cluster k contains 600 observations. In addition, assume there were 30 transitions from cluster i to cluster j, 60 transitions from cluster i to cluster k, and 10 self-returning transitions to cluster i. The corresponding transition probabilities would be: PT_{ik}=0.6, PT_{ij}=0.3, and PT_{ii}=0.1. From the transition probabilities alone, it appears that the transition from cluster i to cluster k is fairly significant. In other words, the transition probabilities point to a relationship between cluster i and cluster k. If the skewness factors are calculated for these transitions, it is found that they are all close to one ($S_{ik} = S_{ij} = S_{ii} = 0.999$). This indicates that if transitions between observations were completely random, approximately the same transition probabilities (PT_{ik}=0.6, PT_{ij}=0.3, and PT_{ii}=0.1) would be obtained for this model. Basically then, contrary to what the transition probabilities suggest for the transitions in question, there is no evidence to indicate that any of them are significant.

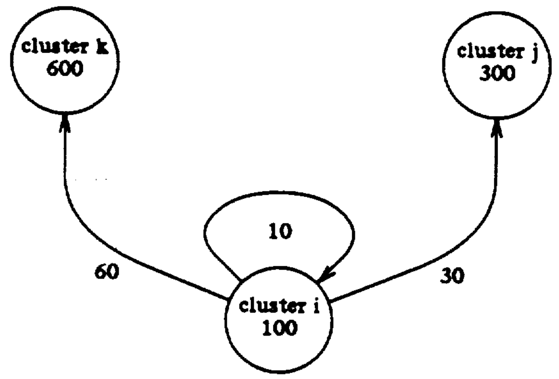

Figure 1
Insignificant Transitions

Therefore, there is no reason to believe that there exists any underlying relationship between the states of the system represented in the cluster model.

5.2. Cluster Analysis for Sample One

The cluster model extracted from Sample One is summarized in Table 7. The means given in this table are calculated from the normalized values of each observation's measurements. They do not, therefore, reflect the actual values obtained, but instead, the values of the observations relative to one another. Observations contained in the first cluster of this model, cluster one, capture a system with many context switches occurring, high IP utilization, and very little user concurrency. Cluster two has similar characteristics, except they are not as extreme. There are fewer context switches and lower IP utilization. The user concurrency is also lower, which leads us to conclude that cluster two depicts the same sort of machine state as cluster one, with fewer jobs running. The observations in these clusters were probably caused by a high degree of multiprogramming which did not allow much concurrency exploitation. Obviously, this is an undesirable state for the parallel computer.

The third cluster, which only accounts for 2.73% of the sample, contains observations with considerable paging activity. As expected, the paging activity is accompanied by above-average IP utilization and context switching. These observations also show extremely low user concurrency.

Table 7
Cluster Means
Sample One

cluster number	% of obs.	context switches	CONCUSER	IPUT	pagact
one	6.01	2.353	-1.113	2.087	-0.135
two	12.02	0.360	-1.281	1.456	-0.135
three	2.73	0.671	-1.211	0.514	4.356
four	26.78	0.356	-0.716	0.124	-0.109
five	30.05	-0.131	0.624	-0.324	-0.135
six	22.40	-1.156	1.152	-1.117	-0.112

The most desirable state, high user concurrency, is captured by the observations found in clusters five, six, and to a lesser degree, cluster four. Cluster six contains observations with much more user concurrency, lower IP utilization, and more paging than the observations in cluster five, which have more concurrency than the observations in cluster four. Thus, state six is more desirable than state five, which is more desirable than state four. It is interesting to note that the high user concurrency captured by observations in these clusters is accompanied by rather low IP utilization and few context switches. This lends credibility to the earlier deduction that the machine was executing in two major states while Sample One was taken (clusters four, five and six depict one state; and clusters one and two depict the other).

All factors considered, the cluster model extracted shows a very efficient environment. The system is in a state of extremely high user concurrency approximately 50% of the time (clusters five and six), with less, but still impressive amounts of concurrency being seen about 25% of the time (cluster four). The undesirable states are contained within clusters one and two, and account for only 18% of the sample.

The transition model extracted for Sample One is shown in Figure 2. The corresponding transition probabilities and skewness factors are shown in Table 8. The high transition probabilities found along the diagonal of this table suggest that for all states (except state three, where the self-returning probability is only 0.2), there is a good chance the machine will operate in the same state during the following observation. The skewness factors confirm this relationship, and show that state three also has an affinity to return to itself. The low transition probability for the self-returning state three transition is caused by the small size of the cluster depicting it.

An interesting phenomenon brought out by the transition model is the lack of interaction between the high and low concurrency states. The only observed transitions into the high user concurrency state (six) were from states four, five, or six, which are other states

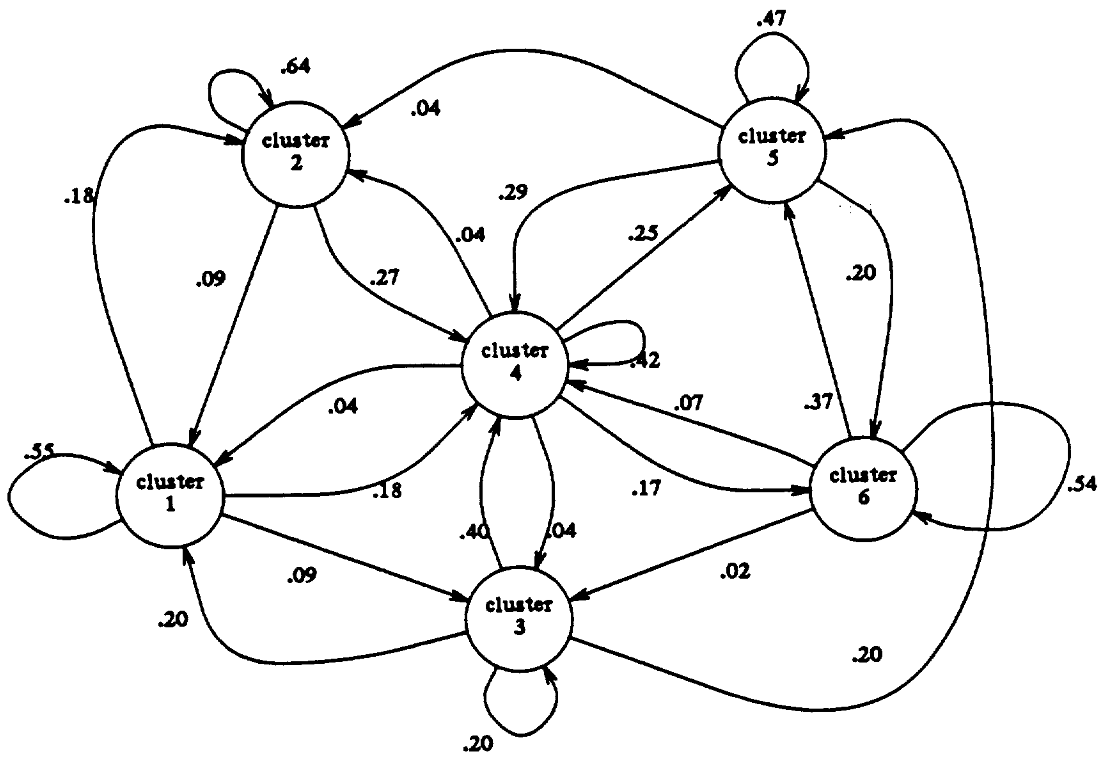

Figure 2
Transition Model
Sample One

depicting substantial user concurrency. This phenomenon is also seen for transitions into state five, the state depicting the second highest degree of concurrency in this model. There are no observed transitions into this state from either of the low concurrency states, one or two. Correspondingly, there are no observed transitions from the two high concurrency states (five and six) into the lowest concurrency state (one). There are also no transitions from state six, and few transitions from state five into state two. In summary, it can be concluded that the machine does not experience sudden jumps from high user concurrency to low user concurrency, or vice versa. Transitions from these extremes are made by stepping through intermediate states, such as state four.

Table 8
Transition Probability/Skewness Factor
Sample One

cluster	one	two	three	four	five	six
one	.5455/9.02	.1818/1.50	.0909/3.31	.1818/0.68	.0000/0.00	.0000/0.00
two	.0909/1.50	.6364/5.26	.0000/0.00	.2727/1.01	.0000/0.00	.0000/0.00
three	.2000/3.31	.0000/0.00	.2000/7.28	.4000/1.49	.2000/0.66	.0000/0.00
four	.0417/0.69	.0833/0.69	.0417/1.52	.4167/1.55	.2500/0.83	.1667/0.74
five	.0000/0.00	.0364/0.30	.0000/0.00	.2909/1.08	.4727/1.56	.2000/0.89
six	.0000/0.00	.0000/0.00	.0244/0.89	.0732/0.27	.3659/1.21	.5366/2.38

The near unity skewness factor for all six transitions from state four indicate that the transitions from this state were almost uniformly distributed among the observations, regardless of the clusters obtained. Obviously, the behavior of the machine after being in this state would be the most difficult to predict. As hinted at above, state four acts as the dispenser, or lowest step, to the extreme states of the system.

A final point of interest is the relationship between state one and state three derived from the skewness factors. The transition probabilities between these states are not very high, but the skewness factors are both 3.31. Recall that both states depict a system of low user concurrency, with state three also corresponding to high paging activity, and state one corresponding to high IP utilization. The explanation for this interstate relationship is rooted in the degree of multiprogramming present on the system.

5.3. Cluster Analysis for Sample Two

A summary of the cluster model extracted for Sample Two is presented in Table 9. The dominant cluster in the model is cluster two which accounts for almost half of the observations. Although the cluster depicts a near idle system, it should not be regarded as a weakness of the machine, but as a consequence of monitoring real workloads. (Long periods of time passed with an extremely light workload while this sample was taken.) For the analysis then, cluster two is ignored when possible, since it reveals little about the system's behavior under a substantial workload. A more revealing cluster, and one which contains observations similar to the undesirable observations found in clusters one and two of the previous model, is cluster one. The observations in this cluster show very little user concurrency, high IP utilization, and a large number of context switches. As before, this behavior is due to system work associated with multiprogramming. Notice, however, that for the first sample the undesirable activity was modeled with two states. Similar modeling is probably being extracted here, with the second undesirable state being hidden in cluster two.

The desirable states, high user concurrency, are captured by the observations in clusters three and five. Cluster three contains observations with higher user concurrency than the observations in cluster five. The very high user concurrency captured by cluster three is accompanied by low IP utilization, little paging activity, and few context switches. Cluster five contains observations with similar, but less extreme, characteristics.

The paging activity that was first discovered in the preliminary analysis is captured by the observations composing clusters four and six. Of the two, cluster six contains the observations with the higher paging activity. The high paging is accompanied by high IP utilization and a large number of context switches. It should also be pointed out that both paging clusters contain observations having little user concurrency, with cluster six (extreme paging observations) showing less concurrency than cluster four (medium paging

Table 9
Cluster Means
Sample Two

cluster number	% of obs.	context switches	CONCUSER	IPUT	pagact
one	7.11	1.335	-0.631	0.272	-0.435
two	49.78	-0.495	-0.794	-0.770	-0.433
three	14.2	-0.708	1.609	-0.124	-0.403
four	12.00	1.290	0.566	1.650	1.566
five	12.44	0.089	1.084	0.846	-0.239
six	4.44	1.945	0.185	1.757	3.277

observations). For both samples, then, paging adversely affected the amount of user concurrency exploited.

If we work under the assumption that cluster two contains only observations of the system under a light workload (which is partially, but not wholly true), we can discard these values for a quick analysis of the efficiency of the system under substantial workload. With the cluster two observations discarded, the percentages of observations for the other clusters are doubled. This puts the system in the desirable clusters (three and five) about 52% of the time, which is similar to Sample One. Continuing with the analysis, we find the system in the paging clusters about 32% of the time, and in the undesirable cluster (one) about 14% of the time. This low percentage of time in the undesirable state is misleading, however, because a number of observations taken during substantial workload were

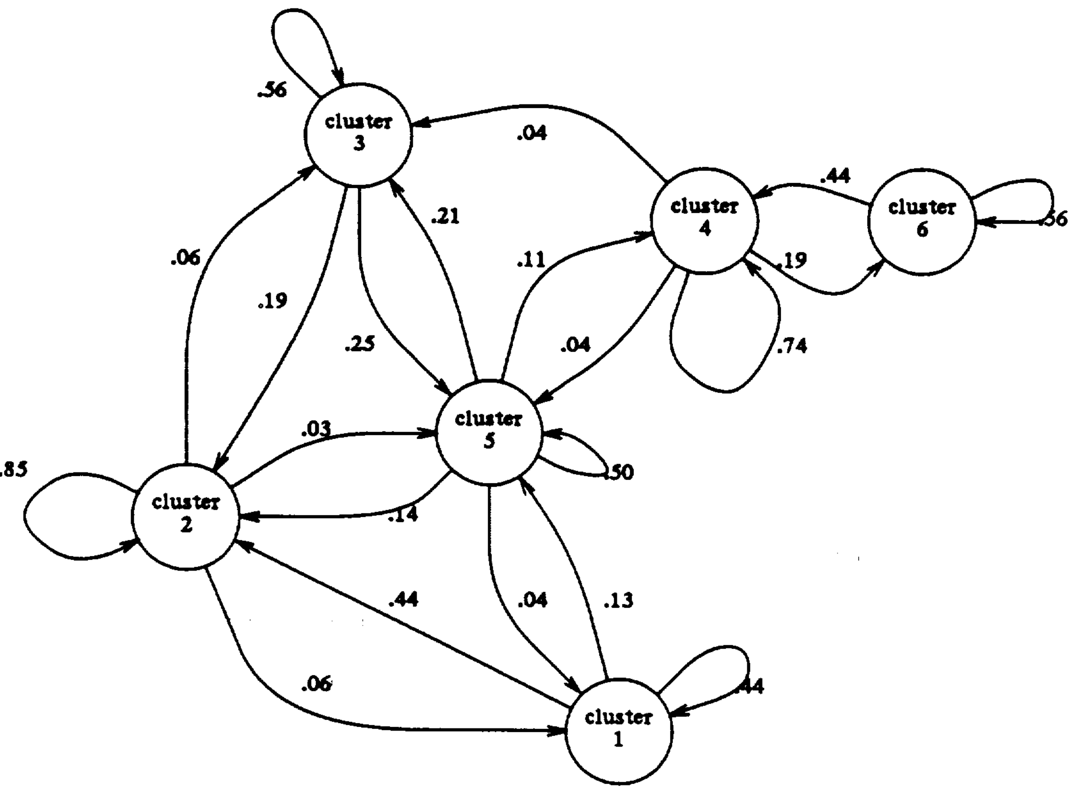

Figure 3
Transition Model
Sample Two

probably discarded with cluster two. In summary, the analysis shows that while the machine was under a substantial workload, which was only about half the time, user concurrency was exploited to high degrees, but not consistently.

The transition model for Sample Two is presented in Figure 3. The corresponding transition probabilities and skewness factors are shown in Table 10. As in Sample One, the transition probabilities and skewness factors are largest for transitions back to the same state (diagonal values in the table). This indicates that the state of the system is fairly steady. The largest of these same state transition probabilities is for cluster two. This reinforces the assumption that long periods of light workload, as depicted by the observations in cluster two, were monitored during this sampling.

The largest skewness factor for a transition from state two, disregarding the self-returning transition, is that for the transition to state one. This, coupled with the fact that the only substantial transition probability into cluster two is that from cluster one, indicates that some observations taken during a substantial workload must be contained in cluster two. This supports the hypothesis that a second undesirable state is hidden in cluster two. This hidden state interacts with the known undesirable state, state one, similarly to the two undesirable states of Sample One.

Table 10
Transition Probability/Skewness Factor
Sample Two

cluster	one	two	three	four	five	six
one	.4375/6.13	.4375/0.88	.0000/0.00	.0000/0.00	.1250/1.00	.0000/0.00
two	.0625/0.88	.8482/1.70	.0625/0.44	.0000/0.00	.0268/0.21	.0000/0.00
three	.0000/0.00	.1875/0.38	.5625/3.94	.0000/0.00	.2500/2.00	.0000/0.00
four	.0000/0.00	.0000/0.00	.0370/0.26	.7407/6.15	.370/0.30	.1852/4.15
five	.0357/0.50	.1429/0.29	.2143/1.50	.1071/0.89	.5000/4.00	.0000/0.00
six	.0000/0.00	.0000/0.00	.0000/0.00	.4444/3.69	.0000/0.00	.5556/12.44

The transition probabilities for the two paging clusters (four and six) are especially interesting because there are few nonzero values. All transitions into or from cluster six come from or go to cluster four or itself. In other words, the only way to get to state six (high paging) was through state four (medium paging), and the only way to leave it was again through state four. This stepping-stone effect goes even further. The only way to get to state four (besides itself or six) was through state five, the third highest state (behind four and six) for paging activity. Therefore, the system gradually builds up to high levels of paging and then gradually dissipates back down to nothing.

As in Sample One, this stepping-stone effect is also seen for the user concurrency measurement. The only tangible (skewness factor > 0.5) path to the state of highest user concurrency (state three) is through state five, which contains observations with the second largest amount of concurrency. If the transition probabilities alone are studied, the path down from high concurrency does not appear to follow the stepping stone routine. There are substantial probabilities for the transitions from the high concurrency states (three and five) to the idle state (two). The low skewness factors for these transitions, prove that these unexpected high transition probabilities are caused by the large size of cluster two. With this information, the stepping stone analogy again makes sense with exits from state three going to state five, and then to state four. In conclusion, as in Sample One, high user concurrency does not come suddenly, but is built up gradually and then gradually dissipates.

CHAPTER SIX

CONCLUSIONS

In this thesis an analysis of an Alliant FX/8 system running Xylem (Cedar's operating system) at the University of Illinois Center for Supercomputing Research and Development was presented. Results for two distinct, real, scientific workloads executing on an Alliant FX/8 were presented. A combination of user concurrency and system overhead measurements was taken for both workloads. Preliminary analysis showed that the first workload sample was comprised of consistently high user concurrency, low system overhead, and little paging. The second sample captured much less user concurrency, but had significant paging and system overhead. In addition, it was determined that both the IPs (interactive processors) and the four computational elements (CEs), while detached, were underutilized.

Statistical cluster analysis was used to extract a state transition model to jointly characterize user concurrency and system overhead. Next, a skewness factor was introduced and used to bring out the effects of unbalanced clustering when determining states with significant transitions.

The results from the models showed that during the collection of the first sample, the system was operating in states of high user concurrency approximately 75% of the time. The second workload sample captured the system in high user concurrency states only 26% of the time. In addition, it was discovered that high system overhead was usually accompanied by low user concurrency. The analysis also showed a high predictability of system behavior, for both workloads. This predictability was largely due to slow changes in system states. In particular, states with extremely high values of paging or user concurrency are usually preceded by states with less paging and user concurrency, much like stair climbing. The opposite effect was observed when the machine left these extreme states.

Future research will include clustering analysis of individual programs and benchmarks to determine their behavior on the system, and to further evaluate the techniques developed. In addition, the same workload will be run under the two operating systems, Xylem and Concentrix, to compare their effectiveness at utilizing the hardware provided. Similar studies on other multiprocessor environments are also in the planning stages.

REFERENCES

[1] E. S. Davidson, D. J. Kuck, D. H. Lawrie, and A. S. Sameh, *Supercomputing Tradeoffs and the Cedar System*, CSRD Report No. 577, University of Illinois at Urbana-Champaign, 1986.

[2] D. J. Kuck, E. S. Davidson, D. H. Lawrie and A. S. Sameh, *Parallel Supercomputing Today and the Cedar Approach*, CSRD Report No. 652, University of Illinois at Urbana-Champaign, June 1987.

[3] P. Yew, *Architecture of the Cedar Supercomputer*, **Proc. IBM Institute of Europe**, pp. 8-12, August 1986.

[4] P. Heidelberger, and K. Trivedi, *Queueing Network Models for Parallel Processing with Asynchronous Tasks*, **IEEE Trans. Comp.**, vol. C-31, pp. 1099-1108, 1982.

[5] P. Heidelberger, and K. Trivedi, *Analytic Queueing Models for Programs with Internal Concurrency*, **IEEE Trans. Comp.**, vol. C-32, pp. 73-82, January 1983.

[6] U. Herzog, W. Hoffman, and W. Kleinoder, *Performance Modeling and Evaluation for Hierarchically Organized Multiprocessor Computer Systems*, **Proc. 1979 Int'l. Conf. on Parallel Processing**, pp. 103-114.

[7] D. J. Kuck, E. S. Davidson, D. H. Lawrie and A. S. Sameh, *The Effects of Program Restructuring, Algorithm Change, and Architecture Choice on Program Performance*, **Proc. 1984 Int'l. Conf. on Parallel Processing**, pp. 129-135.

[8] P. J. McGuire, and R. K. Iyer, *A Measurement-Based Study of Concurrency in a Multiprocessor*, **Proc. 1987 Int'l. Conf. on Parallel Processing**, August 1987.

[9] W. Abu-Sufah and A. Kwok, *Performance Prediction Tools for Cedar: A Multiprocessor Supercomputer*, **Proc. 12th Int'l. Symp. Computer Architecture**, pp. 406-413, 1985.

[10] A. Malony, *Cedar Performance Measurements*, CSRD Report No. 579, University of Illinois at Urbana-Champaign, June 1986.

[11] A. Malony, *Cedar Performance Evaluation Tools: A Status Report*, CSRD Report No. 582, University of Illinois at Urbana-Champaign, July 1986.

[12] M. Devarakonda, and R. K. Iyer, *Predictability of Process Resource Usage: A Measurement-Based Study of Unix*, **Ph.D. dissertation, University of Illinois at Urbana-Champaign**, October 1987.

[13] M. C. Hsueh, R. K. Iyer, and K. Trivedi, *A Measurement-Based Performability Model for a Multiprocessor System*, **Proc. 2nd Int'l. Workshop Applied Mathematics and Performance Reliability Models of Computer/Communication Systems**, May 1987.

[14] D. Ferrari, *On the Foundations of Artificial Workload Design*, ACM, 1984.

[15] Alliant Computer System Corp., *FX/Series Product Summary*, June 1985.

[16] H. Artis, *Workload Characterization using SAS PROC FASTCLUS*, **Proc. Int'l. Workshop Workload Characterization Computer Systems and Computer Networks**, October 1985.

[17] SAS Institute, *SAS User's Guide: Statistics, 1985 Edition.*

[18] SAS Institute, *SAS User's Guide: Basics, 1985 Edition.*